Computers in Peril: A Brief Overview of Metamorphic Viruses

Arvind Kumar Pandey, Assistant Professor

Department of Computer Science, Arka Jain University, Jamshedpur, Jharkhand, India

Email Id- arvind.p@arkajainuniversity.ac.in

ABSTRACT: ***Anti-virus scanners nowadays are generally based on signatures, which search for recognized patterns to determine whether a file is virus-infected. To avoid detection by signature-based scanners, hackers have used code obfuscation techniques to create extremely metamorphic system malware. Signature-based scanners may not be able to identify all of the viruses that exist. Since then, metamorphic malware has evolved from one generation to the next. Hackers employ a variety of methods to attack computers, including metamorphic malware. This article investigates the most prevalent kinds of computer malware and metamorphic computer viruses, as well as the many methods used by metamorphic malware to evade detection. Metamorphic code is code that when run outputs a logically equivalent version of its own code under some interpretation. This is similar to a quine, except that a quine's source code is exactly equivalent to its own output. Metamorphic code also usually outputs machine code and not its own source code.***

KEYWORDS: ***Compute Virus, Malware, Metamorphic Virus, Obfuscation, Polymorphic Virus.***

1. INTRODUCTION

Endpoint security is becoming more important as information technology evolves and improves [1], [2]. A laptop, desktop, server, or mobile device that connects to a network or the internet is an end point. According to Internet Live States, the number of Internet users has increased tenfold between 1999 and 2013, and more than 40% of the world's population now has access to the internet. As a result of the dramatic increase in these statistics in recent years, end point devices must be secured against the massive number of malwares that try to enter such systems through network and internet linked devices. Malicious software, often known as malware, is software that is intended to execute an illegal, frequently dangerous, or undesired action [3]. Malware is a broad word that encompasses a wide range of harmful software, including viruses and worms [4].

The features of metamorphic malware will be examined first in this paper. To do this, we first examine the many kinds of malware that have been created to evade signature-based antivirus scanners, then we go through the numerous sorts of computer viruses and the complexity of metamorphic malware, as well as its features. The parts of this paper are discussed in the following sequence. We'll go through the many kinds of malware in the following part, as well as polymorphic and metamorphic computer viruses, and the intricacy of metamorphic malware. Finally, in the final part, the conclusion will be presented.

2. DISCUSSION

2.1. Common Types of Malware:

The parts that follow go through the many kinds of malware that may be roughly divided into several groups.

2.1.1. Adware:

Adware, sometimes known as advertising sustaining software, is a kind of computer virus that can serve ads on its own [5]. After malicious software or an application is installed or utilized, an adware will show or download advertisements to a computer.

This software is set up to figure out which of the user's favourite websites are and then show ads that are related to their interests. Online free games, peer-to-peer software like torrents, and other adware applications are the most prevalent.

2.1.2. Spyware:

These malwares either spy on or monitor users and collect information about the websites they regularly visit, such as credit card or online banking information, email addresses, and so on [6]. This program allows hackers to get information about a victim's machine without the victim's knowledge or permission. A keylogger program, which is used to monitor the actions of a target machine, is an example of this malware.

2.1.3. Worms:

It's a software that replicates itself over and over again, erasing all data and files on the victim's PC [7]. This software is intended to steal information, destroy files, or infect computers with botnets. Computer worms, according to Cisco, are comparable to viruses and may inflict similar harm. The main distinction is that worms can function as autonomous software and spread on their own, while viruses need human assistance to proliferate. Sending a huge number of emails with infected attachments to a user's contact list is one method of worm distribution.

2.1.4. Trojan Horse:

A Trojan Horse is a malicious program that masquerades as a harmless application [8]. According to Cisco, Trojan viruses cannot re-create itself by infecting another files, nor can they self-replicate. To spread, this kind of malware requires the end user to interact with it, such as downloading and executing a file from the internet or just opening an email attachment. Trojan Horses are categorized in a number of ways depending on how they penetrate systems and how much harm they do. Some of the most popular types are remote access Trojans, proxy Trojans, FTP Trojans, Damaging Trojans, and other Trojan Horses.

2.1.5. Botnet:

Botnet, often referred as bunch of zombies, is a kind of malware that may be used by an attacker to take control of an infected computer or other distant devices [9]. Botnet is a compound word made up of the terms bot and net. Bot is derived from the term "robot" in this context, which typically refers to a PC or laptop infected with malicious software. Net, on the other hand, is derived from the term "network," which refers to a collection of linked computers. Because attackers may not be able to get into individual computers that they have infected while creating malicious software, they use botnets to automatically control a large number of infected machines.

2.1.6. Ransomware:

This is a kind of malicious software that prevents or restricts a user's access to the computer or the data stored on it [10]. These destructors operate by locking either the system's screen or the user's files, with the fraudster demanding a ransom to release them. It's also known as scareware since it scares or intimidates users into paying a charge.

2.1.7. Rootkit:

Another kind of computer malware is a rootkit, which is designed to remotely access a system without being detected by the user or a security software [11].

Because of their covert operation and constant effort to conceal their existence, rootkit attacks may be difficult to avoid and the toughest of all Malwares to detect. As a result, various techniques of detection are used, such as manual detection such as monitoring computer behavior for unusual activity, signature scanning, and so on.

2.1.8. Virus:

A virus is a kind of malicious software that is smaller in size and has the ability to copy itself and propagate to other computers. The most common way for malicious software to propagate is through exchanging software or data across computers. Polymorphic and metamorphic viruses are the two most frequent kinds of viruses, and they will be described in the following section.

2.2. Polymorphic And Metamorphic Computer Viruses:

A virus, as mentioned in the preceding section, is a tiny piece of malicious software with destructive purpose that can quickly replicate itself and spread to other computers. This section is about computer viruses, and it covers two prevalent kinds of viruses: polymorphic and metamorphic viruses.

2.2.1. Polymorphic:

It is one of the most complex kinds of system malware, with the ability to influence many data types and processes [12]. It's a self-encrypting virus with the following characteristics: encryption, self-multiplication, and the capacity to alter one or more components of itself in order to stay undetected. It's designed to escape being detected by a scanner and can make modified duplicates of itself.

As a result, a polymorphic virus has a tendency to mutate in more than one manner before infecting a single machine or a network of computers. Because this dangerous program changes its components correctly and encrypts them, anti-virus systems have a hard time detecting them, and they are considered one of the most clever viruses because they are tough to recognize. Polymorphic viruses may generate an endless number of new decryptors, each of which must employ a different kind of encryption technique to encrypt the virus's constant portion. The development of a polymorphic virus is shown in Figure 1 below.

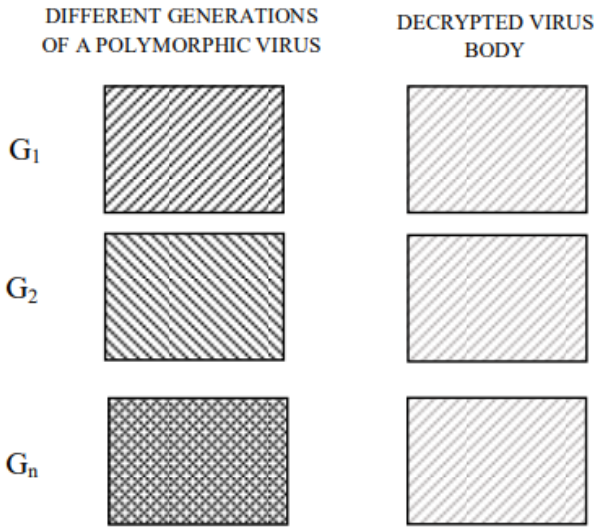

Figure 1: Generation of a polymorphic computer virus

2.2.2. Metamorphic:

A metamorphic virus, according to Kaspersky, is one that may change depending on its capacity to modify, translate, and rewrite its own code. Metamorphic malware is the most contagious kind of malware, and if not discovered early, it may do significant harm to a computer system [13].

Because metamorphic malware has the capacity to alter the underlying structure of the code, reprogram, and rewrite after each infection to a computer system, antivirus systems have a tough time detecting it. To protect computers in networks from contagious metamorphic malware, user administrators should use a multi-layered approach to integrated management that includes a well-defined set of security rules, remote access control limitations, and the deployment of regularly updated antivirus software.

Metamorphism allows a virus to alter its appearance while keeping its functioning. Decryption or encryption methods are not required for the metamorphic virus. On each infection, they show fresh viral bodies. The metamorphic engine may either be included into the virus or kept separate. Figure 2 depicts the generations of a metamorphic virus, whose form changes but not its functioning.

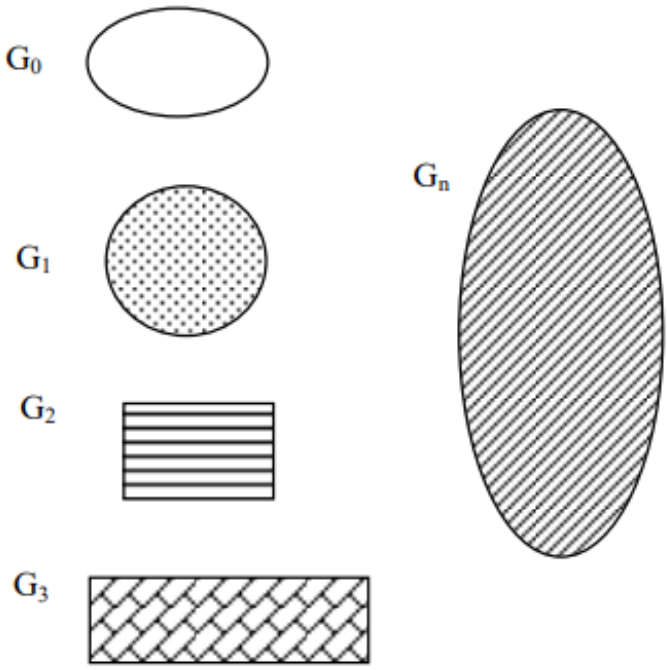

Figure 2: Generation of a metamorphic computer virus [14]**.**

2.3. Obfuscation Techniques:

To create a highly morphing metamorphic virus, virus writers often use a variety of methods. The parts that follow demonstrate the most popular methods for programming a metamorphic virus.

2.3.1. Register Swap Technique:

This is the most basic metamorphic malware approach. In this method, a malware's body is changed by implementing several kinds of registers, despite the fact that the opcodes remain the same between generations. The W95/RegSwap virus is an excellent example of this method. Although the new modified version of the virus is not identical to the old one, the infection's variability is still low, and it may be identified using basic methods such as signature string scanning with a half-byte wildcard. This method is also known as registers renaming or registers exchange. Different generations of RegSwaps are depicted in Figure 3.

a.)

```
5A              pop     edx
BF04000000      mov     edi,004h
8BF5            mov     esi,ebp
B80C000000      mov     eax,000ch
81C288000000    add     edx,0088h
8B1A            mov     ebx, [edx]
899C8618110000  mov     [esi+eax*4+00001118],ebx
```

b.)

```
58              pop     eax
BB04000000      mov     ebx,0004h
8BD5            mov     edx,ebp
BF0C000000      mov     edi,000Ch
81C088000000    add     eax,0088h
8B30            mov     esi, [eax]
89B4BA18110000  mov     [edx+edi*4+00001118],esi
```

Figure 3: Two different generations of RegSwaps [14]

2.3.2. Subroutine Permutation Technique:

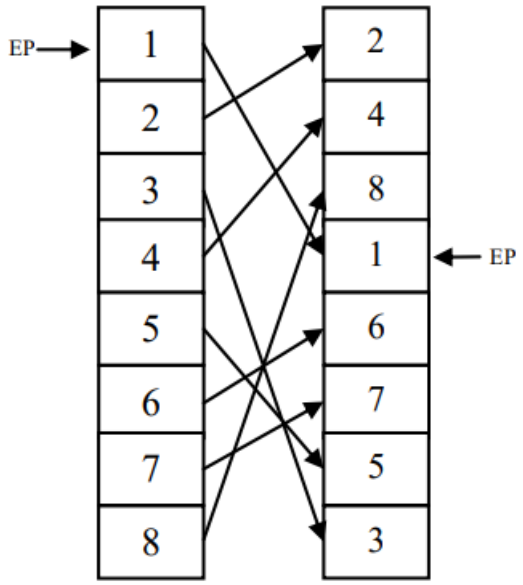

Figure 4: Subroutine permutation Technique [13]**.**

The look of a virus would be altered in this method by rearranging the virus's subroutines. When a virus contains 'n' distinct subroutines, for example, it may produce n factorial different generations without repeating itself. W32/Ghost is an excellent example of a virus that employs this approach. This malicious program includes ten subroutines that can produce ten factorial or 3,628,800 discrete duplicated objects. Nonetheless, since the content of each subroutine remains the same, the infection may be identified using search strings. A Subroutine permutation Technique is depicted in Figure 4 above.

2.3.3. Garbage Instruction Insertion Technique:

To generate morphing copies, certain metamorphic viruses remove and insert junk instruction instructions. This method is sometimes known as "do nothing code," since it does not change the function of an application when run, but it does increase the size of the code.

Viruses using trash instructions are difficult to identify using signatures since this method alters the virus's signature. Within a threshold value, this instruction may be inserted. If the amount of trash instructions is large enough, intrusion detection systems will almost certainly notice the abnormality in the code. As an example of trash instruction codes, consider the table below[12].

3. CONCLUSION

Malware evolution has become a major issue in this decade. Malware is becoming more sophisticated, and it is spreading quicker across global computer networks. Antivirus researchers will have a lot of fun experimenting with new techniques for detecting these destructors now. The Metamorphic malware family is the most difficult threat today since it is very sophisticated, and it has also decreased the value of signature-based detection.

Metamorphic malware is more harder to write for an attacker than polymorphic malware, which requires numerous transformation methods such as register renaming, code reduction, code permutation, and garbage code insertion. As a result, various methods such as generic decryption algorithms, negative heuristic analysis, and others must be used to identify this virus.

In this study, we looked at some of the most prevalent malware kinds, including adware, spyware, worms, Trojan horses, botnets, ransomware, rootkits, and viruses, which may be further categorized. We also looked at various metamorphic malware methods such as register swap, subroutine, and trash instruction, all of which were designed to assist metamorphic malware escape antivirus scanners. The next trend is to do research into artificial intelligence methods in order to develop a more effective method for detecting metamorphic malware.

REFERENCES

[1] Q. Zhu and C. Cen, “A Novel Computer Virus Propagation Model under Security Classification,” *Discret. Dyn. Nat. Soc.*, 2017, doi: 10.1155/2017/8609082.

[2] X. Ma, “Research and Implementation of Computer Data Security Management System,” 2017, doi: 10.1016/j.proeng.2017.01.290.

[3] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. De Geus, “Malicious software classification using transfer learning of ResNet-50 deep neural network,” 2017, doi: 10.1109/ICMLA.2017.00-19.

[4] R. Sihwail, K. Omar, and K. A. Z. Ariffin, “A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, 2018, doi: 10.18517/ijaseit.8.4-2.6827.

[5] S. Yilmaz and S. Zavrak, “Adware: A Review,” *Int. J. Comput. Sci. Inf. Technol.*, 2015.

[6] Symantec employee, “What is spyware? And how to remove it,” *Symantec*. 2017.

[7] B. Rajesh, Y. R. J. Reddy, and B. D. K. Reddy, “A Survey Paper on Malicious Computer Worms,” *Int. J. Adv. Res. Comput. Sci. Technol.*, 2015.

[8] M. A. Orenstein and R. D. Kelemen, “Trojan Horses in EU Foreign Policy,” *J. Common Mark. Stud.*, 2017, doi: 10.1111/jcms.12441.

[9] K. Alieyan, A. Almomani, A. Manasrah, and M. M. Kadhum, “A survey of botnet detection based on DNS,” *Neural Computing and Applications*. 2017, doi: 10.1007/s00521-015-2128-0.

[10] B. A. S. Al-rimy, M. A. Maarof, and S. Z. M. Shaid, “Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions,” *Computers and Security*. 2018, doi: 10.1016/j.cose.2018.01.001.

[11] S. Bhardwaj and K. Pranjal, “Detection Techniques of Rootkits,” *Imp. J. Interdiscip. Res.*, 2016.

[12] I. R. A. Hamid, S. Subramaniam, E. Sutoyo, and Z. Abdullah, “Classification of polymorphic virus based on integrated features,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, 2018, doi: 10.18517/ijaseit.8.6.5045.

[13] D. Lin and M. Stamp, “Hunting for undetectable metamorphic viruses,” *J. Comput. Virol.*, 2011, doi: 10.1007/s11416-010-0148-y.

[14] B. B. Rad and M. Masrom, “Metamorphic Virus Detection in Portable Executables Using Opcodes Statistical Feature,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, 2011, doi: 10.18517/ijaseit.1.4.82.